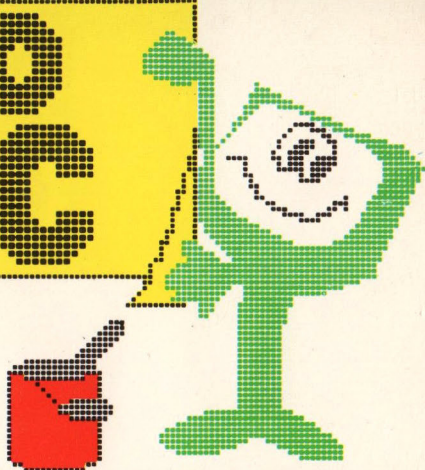
20 VIDEOLEZIONI DI BASIC
PER IMPARARE CON LO SPECTRUM

GRUPPO
EDITORIALE
JACKSON

Indice generale
Indice analitico
Indice alfabetico
Super test

21

Spectrum
16K/48K/PLUS

**VIDEOBASIC SPECTRUM**
Pubblicazione quattordicinale
edita dal Gruppo Editoriale Jackson
**Direttore Responsabile:**
Giampietro Zanga
**Direttore e Coordinatore**
**Editoriale:** Roberto Pancaldi
**Autore:** Softidea -
Via Indipendenza 88-90 - Como
**Redazione software:**
Alessandro Brunetti
Francesco Franceschini,
Luciano Magrini
**Segretaria di Redazione:**
Marta Menegardo
**Progetto grafico:**
Studio Nuovidea - via Loghi, 16 - Milano
**Impaginazione:**
Silvana Corbelli
**Illustrazioni:**
Cinzia Ferrari, Silvano Scolari
**Fotografie:**
Marcello Longhini
**Distribuzione:** SODIP
Via Zuretti, 12 - Milano
**Fotocomposizione:** Lineacomp S.r.l.
Via Rosellini, 12 - Milano
**Stampa:** Grafika '78
Via Trieste, 20 - Pioltello (MI)
**Direzione e Redazione:**
Via Rosellini, 12 - 20124 Milano
Tel. 02/6880951/5


Autorizzazione alla pubblicazione Tribunale di Milano n° 422 del 22-9-1984
Spedizione in abbonamento postale Gruppo II/70 (autorizzazione della Direzione Provinciale delle PPTT di Milano).
Prezzo del fascicolo L. 8.000
Abbonamento comprensivo di 5 raccoglitori L. 165.000
I versamenti vanno indirizzati a: Gruppo Editoriale Jackson S.r.l. - Via Rosellini, 12 20124 Milano, mediante emissione di assegno bancario o cartolina vaglia oppure utilizzando il c.c.p. n° 11666203.
I numeri arretrati possono essere richiesti direttamente all'editore inviando L. 10.000 cdu. mediante assegno bancario o vaglia postale o francobolli.
Non vengono effettuate spedizioni contrassegno.

**Gruppo Editoriale Jackson**

# SOMMARIO



## Introduzione

*Video BASIC ti ha guidato passo dopo passo alla conoscenza del tuo computer trasformandoti in un buon programmatore.*
*A questo punto restano solo i ritocchi conclusivi.*
*Benché Video BASIC sia costituito da venti videolezioni pressoché autonome, i ben tre indici forniti con questo numero (generale, analitico, alfabetico) ti garantiscono una rapida consultazione dell'intera opera e un immediato ripasso degli argomenti da "rinfrescare".*
*Come ogni corso che si rispetti, poi anche Video BASIC si conclude con un esame finale.*
*Ecco allora il Super Test: tante domande per ripassare, verificare, puntualizzare tutto quanto è stato insegnato.*
*In più, abbiamo inserito le doverose correzioni e precisazioni agli ahimé inevitabili errori e refusi che ogni lavoro editoriale comporta, e che, comunque non hanno intaccato minimamente la validità didattica di un'opera come Video BASIC, unica per ricchezza, completezza, facilità.*
*Credendo di aver raggiunto gli obiettivi che vi eravate e ci eravamo proposti, vi ringraziamo per averci sempre seguito.*

*Arrivederci*

# INDICE GENERALE

FASCICOLO 1



FASCICOLO 2

# INDICE GENERALE



FASCICOLO 3



FASCICOLO 4



FASC. 5

# INDICE GENERALE

FASCICOLO 5



FASCICOLO 6



FASCICOLO 7

# INDICE GENERALE

FASCICOLO 7



FASCICOLO 8



FASCICOLO 9

# INDICE GENERALE

FASCICOLO 10



FASCICOLO 11

# INDICE GENERALE

FASCICOLO 11



FASCICOLO 12



FASCICOLO 13



FASC. 14

# INDICE GENERALE

FASCICOLO 14



FASCICOLO 15



FASC. 16

# INDICE GENERALE

**FASC. 16**



**FASCICOLO 17**

# INDICE GENERALE

FASC. 17



FASCICOLO 18



FASCICOLO 19

# INDICE GENERALE

FASCICOLO 20

# INDICE ANALITICO

HARDWARE

# INDICE ANALITICO

HARDWARE

# INDICE ANALITICO

## HARDWARE



## LINGUAGGIO

# INDICE ANALITICO

LINGUAGGIO

# INDICE ANALITICO

LINGUAGGIO

# INDICE ANALITICO

LINGUAGGIO

# INDICE ANALITICO

## LINGUAGGIO



## PROGRAMMAZIONE

# SUPER TEST

NOME E COGNOME ____________________

VIA ____________________ N. ________

CAP ____________ CITTÀ ____________________

ETÀ ____________ PROFESSIONE ____________________

FIRMA ____________________

## Stacca queste 12 pagine Rispondi alle domande del test e spedisci in busta chiusa allegando lire 500 per la risposta al Gruppo Editoriale Jackson Divisione Grandi Opere via Rosellini 12, 20124 Milano

Sono tanti i concetti, le idee, i segreti svelati da Video BASIC in venti videolezioni; sono così indispensabili, per essere un buon programmatore, che è doveroso fare una verifica dell'apprendimento raggiunto. Cosa allora meglio di un Super Test che, spaziando in tutto quanto è stato insegnato, evidenzi i tuoi punti di forza e le tue debolezze, sottolineandoti eventuali lacune o fasi da approfondire?

Se vuoi una verifica alle tue risposte e quindi alle conoscenze acquisite, spedisci il Super Test a Video BASIC: il tuo insegnante personale ti aspetta per l'esame conclusivo.

# SUPER TEST

1) Qual è la principale differenza tra una memoria ROM e una RAM?

☐ A) la ROM è molto meno costosa della RAM
☐ B) la ROM - a differenza della RAM - può essere modificata dal programmatore
☐ C) la RAM - a differenza della ROM - può essere modificata dal programmatore

2) Tra i seguenti computer qualè quello che dispone della più elevata capacità di memorizzazione?

☐ A) 16K ROM, 16K RAM
☐ A) 8K ROM, 32K RAM
☐ B) 32K ROM, 8K RAM

3) Esistono in commercio numerosi computer la cui CPU lavora - anziché su 8 bit - con una capacità di parola di 16 bit.
Indica tra le seguenti risposte quella (o quelle) sicuramente giusta.

☐ A) la CPU a 16 bit è più veloce di quella a 8 bit
☐ B) la CPU a 16 bit esegue operazioni più precise di quella a 8 bit
☐ C) se aumentano i bit, aumenta anche il costo del computer
☐ D) gli elaboratori a 16 bit possono gestire più memoria

4) I bus sono le connessioni che permettono alla CPU di trasportare le informazioni da una parte all'altra del sistema. In qualunque computer ne esistono di 3 distinti tipi:

— bus di controllo
— bus degli indirizzi
— bus ............... Completa la parte mancante.

5) Quanti K di memoria è in grado di indirizzare al massimo un elaboratore che dispone di un bus degli indirizzi a 16 bit?

6) Le porte di Input/Output costituiscono il principale mezzo di comunicazione del computer con il mondo esterno. Tuttavia la connessione non può mai avvenire in modo diretto, bensì mediante un particolare dispositivo, che renda compatibili i segnali sia in ingresso sia in uscita.
Come viene generalmente chiamato tale dispositivo?

# SUPER TEST

7) Qualè la principale differenza tra trasmissione dei dati seriale e trasmissione dei dati parallela?

☐ A) la trasmissione parallela invia un byte alla volta, mentre la seriale un bit alla volta
☐ B) la trasmissione seriale invia un byte alla volta, mentre la parallela un bit alla volta

8) I protocolli di comunicazione, oltre a tutte le norme che vanno rispettate per garantire la compatibiltà di funzionamento, specificano anche la velocità di trasmissione dei dati.
Qual è l'unità di misura abitualmente utilizzata per indicare tale velocità?

9) Il colore delle visualizzazioni sullo schermo di un monitor monocromatico dipende:

☐ A) dal tipo di fosfori utilizzato
☐ B) dal contenuto della memoria colore
☐ C) dall'intensità del fascio elettronico

10) Esistono due tipi di classificazione che vengono normalmente effettuate per catalogare le stampanti: una in base alla velocità di scrittura e l'altra in funzione della qualità di scrittura. Disponi allora il seguente elenco prima in ordine di velocità, quindi in ordine di qualità di scrittura.

— Stampanti ad aghi
— Stampanti a margherita
— Stampanti termiche

11) Quale altro dispositivo di output, oltre alla stampante, permette di eseguire visualizzazioni su carta? Generalmente, questo dispositivo è analogico o digitale? La qualità che esso offre è superiore o inferiore a quella delle stampanti?

12) Supponiamo di conoscere una persona lontana da noi centinaia di chilometri con cui desideriamo entrare in comunicazione attraverso il computer.
Quale dispositivo potremmo utilizzare per eseguire questa operazione senza troppa difficoltà?

# SUPER TEST

13) Oltre alla tastiera, un altro diffusissimo dispositivo di input è costituito dal joystick.
Le informazioni che esso trasmette risultano essere digitali o analogiche? E quelle della paddle?

14) Che cos'è il ciclo di clock?

☐ A) è il tempo necessario per eseguire un'istruzione BASIC
☐ B) è il tempo necessario alla CPU per eseguire una operazione elementare
☐ C) è il tempo necessario per eseguire un ciclo FOR

15) Il tubo catodico è un componente elettronico utilizzato:

☐ A) nel drive
☐ B) nel monitor - TV
☐ C) nella memoria del computer

16) Quando si desidera salvare un programma su nastro magnetico, il computer esegue automaticamente un'operazione preliminare, il cui scopo è quello di memorizzare delle informazioni con cui potrà "sicronizzarsi" nella successiva fase di lettura.
Come viene chiamata in gergo tecnico tale registrazione preliminare?

17) Molti computer sono dotati di discrete capacità musicali, grazie a un generatore sonoro incorporato.
Secondo quale principio vengono di solito generate le forme d'onda che si desidera suonare?

☐ A) analisi
☐ B) sintesi
☐ C) campionamento

18) Può talvolta accadere che nel corso di un trasferimento di informazioni da un'unità o da una zona di memoria ad un'altra (tipicamente tra CPU e periferiche) si renda necessario inserire una memoria di "transito", che compensi le diverse velocità con cui le varie unità possono lavorare.
Come si chiama tecnicamente tale memoria di transito?

# SUPER TEST

19) Il microdrive è un dispositivo ad accesso:

☐ A) casuale
☐ B) sequenziale

20) L'interno del microprocessore del tuo Spectrum può essere immaginato suddiviso in diverse aree, a ciascuna delle quali è assegnato un particolare e specifico compito.

21) Qualè il registro della CPU che si occupa di far svolgere con ordine e continuità logica l'esecuzione dei programmi, indicando cioè di volta in volta la locazione in cui va ricercata l'istruzione da eseguire?

☐ A) il registro di stato
☐ B) l'accumulatore
☐ C) il program counter

## TEORIA

1) Che cos'è il sistema operativo di un elaboratore?

☐ A) l'insieme dei programmi che assicurano il corretto funzionamento di tutta la macchina
☐ B) il gruppo dei programmi che costituiscono nel loro insieme l'interprete BASIC
☐ C) l'insieme dei vari programmi mediante i quali è possibile eseguire le operazioni aritmetiche e logiche

2) Qualè l'equivalente binario del valore decimale 129?

☐ A) 01111111
☐ B) 10000000
☐ C) 10000001

3) Qualè l'equivalente esadecimale del valore decimale 87?

☐ A) 47
☐ B) 57
☐ C) 67

# SUPER TEST

4) Qualè il complemento a due del numero binario 01101101?

☐ A) 10101001
☐ B) 10010011
☐ C) 00010011

5) Qualè il risultato della seguente sottrazione tra valori esadecimali: A1 - 2E?

☐ A) 73
☐ B) 7A
☐ C) 8F

6) Qualè l'operazione che si deve eseguire su una cartuccia vergine per microdrive prima di potervi registrare qualsiasi informazione?

7) Qualè la principale caratteristica che differenzia un linguaggio interpretato rispetto a un linguaggio compilato?

☐ A) La velocità di esecuzione
☐ B) Il numero di funzioni disponibili
☐ C) La diversa disposizione delle variabili in memoria

8) Che cos'è un linguaggio di programmazione ad alto livello?

☐ A) Un linguaggio molto più vicino alla macchina che all'uomo
☐ B) Un linguaggio molto più vicino all'uomo che alla macchina
☐ C) Un linguaggio vicino sia all'uomo che alla macchina

9) Scrivi almeno un esempio di linguaggio ad alto livello e uno di linguaggio a basso livello.

10) Qualè il termine tecnico usato normalmente per indicare la fusione tra due file (cioè il raggruppamento di due distinti file in uno unico)?

11) Qualè la forma in notazione scientifica del valore 1200000?

☐ A) 12E5
☐ B) 0.12E7
☐ C) 1.2E6

12) Quanto valgono mantissa ed esponente di 1.23E-4?

- ☐ A) 1.23 e -4
- ☐ B) 1.23 e 4
- ☐ C) 1 e 4

13) Scrivi a fianco di ciascuna delle seguenti istruzioni Assembler il modo di indirizzamento che è stato utilizzato.

a) LD A, C..........................................................................
b) LD A, (IY+3) ................................................................
c) RRA ...............................................................................
d) LD A, BBH.....................................................................

14) Esiste un codice internazionale in base al quale molti elaboratori convertono in particolari configurazioni binarie tutti i caratteri standard dell'alfabeto, i numeri e numerosi codici di controllo.
Tale codice viene diligentemente applicato anche dal tuo computer: qualè la sua sigla?

15) Qualè la differenza (o le differenze) tra codice oggetto e codice sorgente?

- ☐ A) l'oggetto è ciò che si ottiene dopo la compilazione del sorgente
- ☐ B) il sorgente è ciò che si ottiene dopo la compilazione dell'oggetto
- ☐ C) il sorgente è listabile e correggibile, al contrario dell'oggetto
- ☐ D) l'oggetto è listabile e correggibile, al contrario del sorgente.

## LINGUAGGIO

1) Qualè il comando BASIC che permette di assicurarsi che un programma sia stato registrato correttamente sul supporto magnetico?

- ☐ a) LOAD
- ☐ b) SAVE
- ☐ c) VERIFY

# SUPER TEST

2) Qualè la funzione che permette di conoscere la situazione degli attributi in una certa posizione dello schermo video?

3) Qualè l'operatore che permette di eseguire la concatenazione tra due o più stringhe?

☐ A) + ☐ B) − ☐ C) *

4) Scrivi il risultato che apparirebbe sullo schermo eseguendo questa istruzione:

```
PRINT CHR$ (CODE "BASIC")
```

5) Qualè l'istruzione che si deve impartire prima di poter utilizzare un array?

☐ A) LET ☐ B) DIM ☐ C) TAB

6) L'esecuzione di questo programma continuerà alla linea 30 oppure salterà alla riga 100?

```
..................
10 LET A=6: LET B=4: LET C=3.5
20 IF A>B AND (B<C OR A>C) THEN GOTO 100
30 ..............
```

7) Se nel seguente programma:

```
............
10 INPUT A$
20 PRINT A$
30 STOP
```

volessimo evitare di battere RETURN dopo aver impostato il valore di A$, come si dovrebbe modificare la linea 10?

8) Scrivi le due istruzioni necessarie per memorizzare uno stesso programma (chiamiamolo "PROVA") sia sul registratore che sul microdrive.

9) Quali sono le istruzioni Assembler equivalenti all'istruzione BASIC
LET A=3:POKE 4000,A

# SUPER TEST

10) Così come PRINT è il comando per visualizzare qualcosa sullo schermo, qualè l'istruzione che ordina la scrittura sulla stampante?

11) Definisci una funzione il cui risultato sia pari all'argomento modificato per 3 e diviso per 40.

12) Qualè il risultato di PRINT 27 AND 90?

13) Scrivi a fianco di ciascuna delle seguenti istruzioni il rispettivo risultato:

PRINT INT (-1.23)
PRINT ABS (-1.23)
PRINT SGN (-1.23)
PRINT ABS( INT (-1.23))
PRINT INT( ABS (-1.23))

14) Secondo quale priorità vengono elaborate le istruzioni che contengono gli operatori logici AND, OR e NOT?

☐ a) OR, NOT, AND
☐ b) AND, OR, NOT
☐ c) NOT, AND, OR

15) Scrivi le istruzioni necessarie per riuscire ad estrarre dalla stringa "RIVISTA" le seguenti sottostringhe:

a) RIV

b) VIS

c) VISTA

16) Quale istruzione deve essere eseguita prima di poter utilizzare il comando PRINT#?

☐ a) CLOSE# ☐ b) INPUT# ☐ c) OPEN#

# SUPER TEST

18) Qualè l'istruzione mediante la quale è possibile resettare tutte le variabili - sia di tipo stringa che numerico - senza però perdere il programma in memoria?

☐ a) NEW
☐ b) CLEAR
☐ c) CLS

19) Specifica a fianco di ciascuna delle seguenti istruzioni il rispettivo risultato. Supponi che B$="BUON" e C$="GIORNO"

a) PRINT LEN (C$+B$)
b) PRINT LEN (C$)+LEN(B$)
c) PRINT LEN (B$)-LEN(C$)
d) PRINT LEN (B$-C$)

20) Dato il seguente programma:

```
10 RANDOMIZE
20 INPUT A
30 PRINT INT(RND*A)
```

☐ a) 1020
☐ b) 1021
☐ c) 1022

quale valore devi assegnare alla variabile A per ottenere la stampa di un numero sicuramente compreso tra 0 e 1021?

21) Quanti elementi contiene l'array definito da DIM C(4,2)?

☐ a) 8 ☐ b) 12 ☐ c) 15

22) Se A$ = "1A2B3C4D", cosa producono le seguenti istruzioni?

a) A$ (1 TO 3)
b) A$ (3 TO 6)
c) A$ (7 TO 8)

23) Questo programma somma 1/3 per 3000 volte:

```
10 LET A=1/3: LET SOMMA=0
20 FOR I=1 TO 3000
30 LET SOMMA=SOMMA+A
40 NEXT I
50 PRINT SOMMA
```

Perché al termine dell'esecuzione la stampa della variabile SOMMA non risulterà corretta? (Il risultato dovrebbe essere infatti pari a 3000*1/3=1000)

# SUPER TEST

## PROGRAMMAZIONE

1) Quante volte viene eseguito questo ciclo FOR?

```
10 FOR I=0 TO 1124
20 PRINT "*";
30 NEXT I
```

☐ a) 0 ☐ b) 1124 ☐ c) 1125

2) Quale dovrebbe essere il contenuto delle variabili A$, B$ e C$ dopo che questo segmento di programma ha letto i valori nelle DATA?
Il programma funziona correttamente? Perché?

```
10 READ A$
20 READ B$
30 READ C$
40 GOTO 30
50 DATA "CARTA","PENNA","GOMMA"
60 DATA "MATITA","TEMPERINO"
70 DATA "CALAMAIO"
```

3) Questo programma - apparentemente corretto - provoca invece la visualizzazione di un messaggio di errore: per evitarlo inserisci l'istruzione necessaria.

```
 10 INPUT A
 20 INPUT B
 30 GOSUB 100
 40 LET SOMMA=A+B
 50 PRINT SOMMA
100 PRINT "A=";A
110 PRINT "B=";B
120 RETURN
```

4) La seguente riga di programma:

```
10 FOR I=1 TO 100: PRINT I: NEXT I
```

provoca la stampa sullo schermo dei primi 100 numeri interi.
Specifica quale modifica si rende necessaria perché di tali numeri vengano visualizzati soltanto quelli dispari.

# SUPER TEST

5) Supponi di avere un programma che contenga - tra le altre - anche questa linea: 85 IF A$<B$ THEN STOP
Scrivi i valori di una possibile coppia di stringhe che, inserite rispettivamente in A$ B$ e provocherebbero l'arresto dell'esecuzione.

6) Come scriveresti un'istruzione BASIC per ribaltare, uno rispetto all'altro, i due gruppi di 4 caratteri di una stringa lunga complessivamente 8 caratteri (come per esempio "PIRAMIDE" → "MI-DEPIRA")?

7) Il risultato di queste linee di programma risulta essere la stampa del contenuto dei vari elementi che appartengono all'array N, a cominciare dal primo.

```
50 FOR I=1 TO 100
60 PRINT N(I)
70 NEXT I
```

In quale modo va modificata la linea 50 perché la visualizzazione avvenga invece partendo dall'elemento numero 100?

8) Quale linea di questo programma non verrà mai eseguita?

```
10 PRINT "DI QUI PASSO DI SICURO"
20 INPUT B
30 IF B>0 THEN GOSUB 70
40 PRINT "ANCHE DI QUI PASSO DI SICURO"
50 STOP
60 RETURN
70 PRINT "QUESTA STAMPA NON È CERTA"
80 RETURN
```

9) Quale errore è stato commesso in questo programma per la stampa della tavola pitagorica?

```
10 REM TAVOLA PITAGORICA
20 FOR A=1 TO 10
30 FOR B=1 TO 10
40 PRINT A;"*";B;"=";A*B
50 NEXT A
60 NEXT B
```

10) Scrivi un programma che assegni alla variabile A il valore 3, alla variabile B il valore 12 e quindi esegua lo scambio dei valori contenuti in A e B (suggerimento: è necessario ricorrere ad una terza variabile per ....)

# INDICE ANALITICO

PROGRAMMAZIONE

# INDICE ALFABETICO

# INDICE ALFABETICO

# INDICE ALFABETICO

# INDICE ALFABETICO

# INDICE ALFABETICO

# INDICE ALFABETICO

# INDICE ALFABETICO

# INDICE ALFABETICO

# INDICE ALFABETICO

# INDICE ALFABETICO

# MODIFICHE

**Fascicolo 1 a pag. 27**
**modifica così:**

## La sintassi del comando LET

LET variabile = espressione

## LINGUAGGIO

| | |
|---|---|
| PRINT 12;34 | Visualizza 1234 cioè stampa consecutivamente due costanti numeriche; |
| PRINT 12;—34 | Stampa consecutivamente le costanti 12 e —34; |
| PRINT 54; " "; 96; | Stampa la costante 54, poi uno spazio, poi la costante 96 e predispone la stampa della successiva istruzione PRINT subito dopo il 96; |
| PRINT "JACK"; "SON" | Stampa una di seguito all'altra le due costanti testo. |
| LET N$="VIDEO" LET T$="BASIC" PRINT M$; B$ | Stampa uno di seguito all'altro i valori delle variabili testo N$ e T$, cioè VIDEOBASIC. |
| LET AFFITTO=450000 LET TELEFONO=96500 PRINT AFFITTO, TELEFONO | Incolonna i valori delle variabili, stampando quello di AFFITTO all'inizio della linea e quello di TELEFONO a partire dalla colonna 16 della stessa riga. |
| PRINT, "PAPER SOFT" | Stampa la costante testo PAPER SOFT a partire dalla colonna 16. |

È possibile stampare i valori delle variabili, se questi sono stati precedentemente assegnati:

**Sintassi dell'istruzione PRINT**

PRINT [espressione] [{;} espressione]

29

**Fascicolo 1 a pag. 29**
**modifica così:**

```
LET N$ = "VIDEO"
LET T$ = "BASIC"
PRINT N$; T$
```

## VIDEOESERCIZI

Annota, nello spazio apposito, il risultato da te previsto per ciascun esercizio proposto e poi verificalo con la soluzione del tuo Spectrum. Se avrai commesso anche un solo errore ripassa la lezione. Descrivi il risultato di stampa sul video delle seguenti istruzioni.

PRINT "Prova 1": PRINT "Prova 2"

PRINT "Prova 3";: PRINT "Prova 4"

PRINT "Prova 5",: PRINT "Prova 6"

PRINT "Prova 7": PRINT: PRINT "Prova 8"

PRINT, "FINE"

Introduci ora le istruzioni alla tastiera e confronta i risultati. Scrivi di fianco le informazioni che pensi vengano visualizzate dal tuo computer dopo aver introdotto le seguenti istruzioni:

PRINT 110-20 PRINT "110-20"

PRINT 3.5 PRINT 3.5

PRINT 3,5

LET A1=30: LET A1=—10: PRINT A1

LET X=12: LET Y=13: PRINT X+Y

PRINT "X+Y"

LET P=100: LET N=10: LET T=P+N: PRINT T

32

**Fascicolo 1 a pag. 32**
**modifica così:**

```
PRINT 110 – 20 :
PRINT "110 – 20"
PRINT 3.5 : PRINT 3,5
```

## Sintassi dell'istruzione PRINT

PRINT [ espressione ] [ { ; , ' } espressione ]

# MODIFICHE

**Fascicolo 2 a pag. 19 modifica così:**

## Sintassi dell'istruzione

INPUT [ "Messaggio"; ] variabile [ { ' ; , } variabile ... ]

**Fascicolo 3 a pag. 28 modifica così il Flow chart:**

LINGUAGGIO

**CHR$**

CHR$ è, in un certo senso, la funzione opposta di CODE: essa ti restituisce il carattere corrispondente al numero che avrai utilizzato come argomento.

Tale numero può essere specificato sia per mezzo di una variabile che di un'espressione. L'argomento deve essere compreso tra 0 e 255; in caso contrario l'elaboratore visualizzerà il messaggio di errore INTEGER OUT OF RANGE.
Quando il valore dell'argomento è un numero decimale, CHR$ lo tronca al valore intero immediatamente inferiore:

```
15 PRINT CHR$(151/2)
```

151/2 = 75.5: il numero intero immediatamente più piccolo è 75. A tale argomento corrisponde il carattere K.
Il seguente programma stampa l'intero set dei caratteri disponibili sul tuo Spectrum:

```
10 FOR A = 0 TO 255: PRINT CHR$ (A): NEXT A
```

**Fascicolo 5 a pag. 12 modifica così:**

lo arrotonda al valore intero più vicino:

```
10 FOR A = 32 TO 255 :
   PRINT CHR$ (A) :
   NEXT A
```

**Fascicolo 4 a pag. 31 modifica così la linea 70 del listato:**

```
70 LET SOMMA = SOMMA + N : LET NUM =
   NUM + 1 : GOTO 40
```

**Fascicolo 5 a pag. 16 modifica così:**

```
10 LET A$ = INKEY$
20 IF A$ = ""
   THEN GOTO 10
30 PRINT A$
```

# MODIFICHE

## LINGUAGGIO

PAUSE gode inoltre di un'ulteriore particolarità: se nel corso della sosta viene premuto un tasto qualunque, il programma riprende immediatamente a lavorare, abbreviando la pausa prevista in origine. Questo comando può essere quindi utilmente adoperato per adeguare il tempo di permanenza sullo schermo delle varie visualizzazioni alle capacità di lettura dell'utente del programma.

### Sintassi del comando

PAUSE n
dove n è un valore numerico compreso tra 0 e 65535.

### FOR, TO, STEP, NEXT

Accade spesso che un programma richieda di eseguire più volte un'istruzione od un gruppo di istruzioni. Supponi, a titolo di esempio, di voler scrivere un programma che moltiplichi la variabile A per i valori 1, 2, 3, 4, e 5. Una possibile soluzione potrebbe essere:

```
10 LET I = 1 : REM I è
   il numero
   da moltiplicare
   per A
20 LET A = A * I
30 LET I = I + 1
40 IF I < 6 GOTO 20
```

Si tratta di un tipico esempio di ciclo, cioè una sequenza di istruzioni eseguita un certo numero di volte. Esiste in BASIC una istruzione particolare che ti permette di realizzare i cicli senza ricorrere a strutture complesse. L'istruzione è composta dalle parole FOR...NEXT.
Immagina di dover ripetere una serie di istruzioni per un determinato numero di volte, e precisamente finché un certo contatore C (avente inizialmente il valore S) raggiunga il valore A. Utilizzando l'istruzione FOR...NEXT potrai scrivere:

```
FOR C = S TO A
```

fai qualcosa

19

**Fascicolo 5 a pag. 19**
**modifica così:**

```
10 LET I = 1 : REM I è
   il numero
   da moltiplicare
   per A
20 PRINT I * 2
30 LET I = I + 1
40 IF I < 6 GOTO 20
```

## LINGUAGGIO

```
NEXT C
```

La prima volta che il ciclo viene eseguito C è posto uguale al valore di S. Si eseguono quindi tutte le istruzioni.
Arrivati a NEXT il valore di C viene incrementato e confrontato automaticamente con A. Se C risulta minore di A, il ciclo viene nuovamente ripetuto, altrimenti si continua con l'istruzione che segue il NEXT.
La procedura prosegue pertanto finché il contatore C raggiunge il valore finale di A.
Riprendendo il problema della moltiplicazione si poteva allora dare questa soluzione:

```
10 FOR C = 1 TO 5
20 LET A = A * C
30 PRINT A
40 NEXT C
```

C si chiama variabile di controllo del ciclo (o contatore) e può assumere un qualsiasi nome (legale) consentito. Nell'esempio avrai notato che 1 è il suo valore iniziale e 5 è il valore finale o di test.
Nell'istruzione FOR si possono utilizzare anche delle espressioni, per esempio:

```
FOR A = M + 5 TO B/5
```

oppure delle variabili, a patto che siano prima



**Fascicolo 5 a pag. 20**
**modifica così:**

```
10 FOR C = 1 TO 5
20 PRINT A * C
30 NEXT C
```

## LINGUAGGIO

state impostate. Ad esempio:

```
10 LET DA = 5 : LET A = 15
20 FOR C = DA TO A
30 ...
40 NEXT C
```

È inoltre possibile incrementare il valore della variabile contatore con un passo diverso da 1, semplicemente utilizzando la parola STEP (passo) seguita dal valore di cui si vuole incrementare ogni volta il contatore:

```
FOR I = 2 TO 10 STEP 2
```

I assumerà allora i valori 2, 4, 6, 8, 10.
È anche possibile scrivere programmi in cui i cicli contengano al proprio interno altri cicli:

```
10 FOR I=0 TO 10
   STEP 3
20 FOR J=3 TO 9
30 —
40 —
50 —
60 NEXT J
70 NEXT I
```

Si dice allora che i cicli sono nidificati (ti ricordi di IF...THEN GOTO...). Naturalmente i contatori dei singoli cicli devono essere diversi. Il ciclo interno ad un altro ciclo deve inoltre essere completamente contenuto nel primo ciclo; non è quindi possibile sovrapporre parti di cicli. Ad esempio:

```
100 FOR I=13 TO 20
110 FOR J=0 TO 10
120 —
130 —
140 —
150 —
160 NEXT I
170 NEXT J
```

È sbagliato! I due cicli sono infatti sovrapposti. Molto pericoloso!
Se la variabile indice è inizialmente più grande del valore finale, il ciclo viene eseguito una sola volta. Ad esempio:

21

**Fascicolo 5 a pag. 21**
**modifica così:**

del valore finale, il ciclo non viene eseguito.
Ad esempio: ecc...

**Fascicolo 5 a pag. 31**
**modifica così la linea 20**
**del listato:**

```
20 IF NUM<>INT (NUM) THEN GOTO 5 : REM ecc....
```

**Fascicolo 7 a pag. 15**
**modifica così il commento**
**al secondo esempio:**

Per A = 0 (infatti NOT 0 = 1)
o B <> 0, M
assumerà il

# MODIFICHE

VIDEOESERCIZI

Annota nello spazio apposito il risultato da te previsto per ciascun esercizio proposto e poi verificalo con la soluzione del tuo computer. Se avrai commesso anche un solo errore, ripassa la lezione.

```
10 LET A = 12 : LET B = 5 : LET N$ = "BENE"
20 IF A < B THEN PRINT "MELE"
30 IF B < A THEN PRINT "BANANE"
40 IF (A + B) <> (B + A) THEN PRINT N$
```

Quale è la materia considerata più importante in questo listato?

```
10 CLS : PRINT "CHE VOTO MERITI?"
20 INPUT "TEORIA"; T
30 INPUT "PROGRAMMAZIONE"; P
40 LET M = T + P/2
50 IF P >= 7 AND M > 6 THEN PRINT "PROMOSSO"
60 IF P < 7 OR M <= 6 THEN PRINT "RESPINTO"
```

Secondo te, con 3 in teoria e 8 in programmazione si è promossi?

Quante volte verrà eseguita l'elaborazione di questo Loop?

```
10 FOR I = 1 TO 10
20 PRINT I : CLEAR
30 NEXT I
```

Perché?

**Per chi ama nascondere i propri segreti ...**

```
10 PAPER 2 : INK 2: CLS
20 PRINT AT 4, 4; "È UN SEGRETO"
30 PAUSE 100
40 INK 7 : PRINT AT 10,4; "NON È UN SEGRETO"
```

Una tecnica usata per nascondere i listati da sguardi indiscreti è quella di stamparli con inchiostro uguale al colore dello sfondo (carta).

32

**Fascicolo 7 a pag. 30 modifica così:**

```
30 IF A >= B AND A >= C AND (NOT B) >= C ecc....
```

**Fascicolo 7 a pag. 32 modifica così la linea 40 del secondo videoesercizio:**

```
40 LET M = (T + P)/2
```

**Fascicolo 8 a pag. 13 modifica così l'ultimo esempio:**

```
LET A (13) = 5.3
```

LINGUAGGIO

```
F(19) = F(19) - 3.17
```

Diminuisce di 3.17 il valore contenuto nella diciannovesima variabile del vettore F.

```
C$ (3) = "marzo"
```

Assegna al terzo elemento dell'array C$ (di tipo stringa) il valore "marzo".

È possibile, anziché utilizzare un numero, specificare l'indice mediante una variabile od un'espressione. In tal caso si dovrà fare particolare attenzione che in nessuna circostanza questa variabile o questa espressione assuma valori non leciti (maggiore della dimensione massima dell'array o minore di 1).

```
LET K = 7
LET V(K) = 73
```

Assegna al settimo elemento del vettore V il valore 73

```
LET V (K - 13) = 9
```

Errore! Fuori dai limiti! La possibilità di rappresentare l'indice con variabili consente per esempio di scandire tutto un array, utilizzando un ciclo:

```
10 DIM V$ (7)
20 LET V$ (1 TO 7) =
   "VACANZE"
 90 ......
110 FOR I = 1 TO 7
120 PRINT I V$ (I)
130 NEXT I
140 ......
```

14

**Fascicolo 8 a pag. 14 modifica così:**

```
LET F (19) = F (19) - 3.17
```

```
LET C$ (3) = "marzo"
```

**Fascicolo 8 a pag. 23 modifica così:**

## Sintassi dell'istruzione/comando

LOAD "[ nome programma ]"

# MODIFICHE

**Fascicolo 8 a pag. 24 modifica così:**

## Sintassi dell'istruzione

VERIFY "[ nome del programma ]"

## Sintassi dell'istruzione

MERGE "[ nome del programma ]"

LINGUAGGIO

**Esempi**

| | |
|---|---|
| LET A$ = STR$ 50 | Assegna ad A$ il valore alfanumerico "50". |
| PRINT STR$ (3 * 7 − 2) | Risultato: 19. |
| PRINT VAL STR$ 50 | È un'istruzione inutile, per quanto perfettamente lecita. Il valore numerico 50 viene infatti prima convertito in stringa (con STR$) e quindi ritrasformato in numero. Sullo schermo comparirà allora 50, inteso come numero, non come stringa. |

**Sintassi della funzione**

STR$ (espressione)

**TO**

L'operazione di concatenamento permette di aggiungere caratteri ad una stringa (ricordandosi comunque di non superare il limite massimo di 255 caratteri) eseguendo cioè una sorta di somma tra stringhe.
In certi casi, anziché aggiungere, potrebbe invece essere necessario dover estrarre dei caratteri da una stringa: esiste allora nel linguaggio BASIC del tuo Spectrum una funzione mediante la quale è possibile eseguire questa operazione di separazione: TO.
Il risultato di TO è una sottostringa (cioè un gruppo di caratteri appartenenti alla stringa originale) estratta a partire da una certa posizione specificata e lunga tanti caratteri quanti ne compaiono fino alla posizione dove si vuole che la sottostringa termini.
Vediamo subito un esempio chiarificatore:

PRINT "GIAN CARLO" (1 TO 4)

Stampa quattro caratteri della stringa "GIAN

15

**Fascicolo 9 a pag. 15 modifica così:**

L'operazione di concatenamento permette di aggiungere caratteri a una stringa, eseguendo cioè una sorta di somma tra stringhe.

PROGRAMMAZIONE

**Uso di Vettori e matrici**

Il primo programma che oggi esamineremo è un esempio introduttivo all'uso degli array. Adoperando i vettori è infatti indispensabile avere ben chiari nella mente i concetti di elemento, valore ed indice: questo programma dovrebbe perciò aiutarti a risolvere eventuali dubbi ed incertezze.
Ciò che ci proponiamo di fare è molto semplice: definire un array di 7 elementi e memorizzare in ciascuno di essi un valore a caso - compreso tra 0 e 100 - generato dalla funzione RND.



**Fascicolo 8 a pag. 26 modifica così:**

compreso tra 0 e 99 - generato dalla funzione RND.

# MODIFICHE

**Fascicolo 10 a pag. 18**
**modifica così:**

## Sintassi dell'istruzione

PRINT # numero { ; , ' } dati

**Fascicolo 10 a pag. 23**
**modifica così:**

## Sintassi dell'istruzione

LPRINT espressione [ { ' , ; } espressione ]

LINGUAGGIO

**Esempi**

| | |
|---|---|
| CIRCLE 127, 87, 87 | Disegna il più grosso cerchio che riesce ad essere interamente contenuto nello schermo. |
| CIRCLE 100, 70, 96<br>CIRCLE 100, 70, 48 | Disegna due circonferenze concentriche, una di raggio doppio dell'altro. |
| 10 CLS<br>20 LET X = INT (RND * 235) + 10<br>30 LET Y = INT (RND * 155) + 10<br>40 LET COLORE = INT (RND * 7)<br>50 INK COLORE<br>60 CIRCLE X, Y, 10<br>70 GOTO 20 | Vengono tracciate circonferenze di centro e colore casuale, ma tutte con raggio 10. |

**Sintassi del comando**

CIRCLE coordinata X, coordinata Y, raggio

24

**Fascicolo 12 a pag. 24**
**modifica così:**

```
CIRCLE 100, 70, 70
CIRCLE 100, 70, 35
```

**Fascicolo 12 a pag. 31**
**modifica così la linea 30**
**del listato:**

```
30 FOR I = 1 TO P
```

**Fascicolo 12 a pag. 32**
**modifica così la linea 10**
**del primo videoesercizio**

```
10 LET A$ = "INT RND * 6" : REM INT e RND sono
   parole chiave digitate in modo esteso e non carattere
   per carattere
```

# MODIFICHE

**Fascicolo 13 a pag. 19 modifica così la linea 500 del listato:**

```
500 FOR I = 1 TO LEN (C$) + 4
```

LINGUAGGIO

operare.
Arriviamo infine alle ultime due considerazioni che, per quanto all'apparenza banali, possono in numerose circostanze evitare contrattempi ed errori apparentemente inspiegabili.

La prima riguarda la collocazione delle subroutine: è sempre bene metterle in fondo al programma principale, numerandole - come nel nostro caso - in maniera che risaltino immediatamente alla prima occhiata. In casi di correzione, questo ti eviterà sicuramente errori, incertezze e perdite di tempo.
La seconda osservazione riguarda invece l'istruzione STOP che, come puoi vedere, è stata messa soltanto nella seconda versione: nella prima - nonostante fosse perfettamente lecito - ha potuto essere trascurata, dal momento che, quale ultima istruzione si poteva tralasciare.
Prova adesso a toglierla alla seconda versione e ad eseguire il programma: vedrai apparire sullo schermo il messaggio di errore RETURN WITHOUT GOSUB.
Ciò è dovuto al fatto che lo Spectrum non sa dove termina il programma vero e proprio: invece di "capire" che l'esecuzione va arrestata subito dopo la linea 100, esso prosegue nelle linee successive, trattandole come se facessero parte del corpo principale e non della subroutine.
Il RETURN alla riga 540, non essendo stato chiamato da alcun GOSUB, provoca quindi l'insorgere dell'errore.
Il modo migliore per separare il programma principale dalle subroutine è allora quello di inserire l'istruzione STOP: in questa maniera ci si mette sicuramente al riparo da pericolose eventualità di errori od arresti indesiderati.

**Sintassi dell'istruzione GOSUB**

GOSUB espressione numerica

**Sintassi del comando RETURN**

Return

21

**Fascicolo 13 a pag. 21 modifica così:**

subroutine: è sempre bene metterle in fondo o all'inizio del programma principale.

**Fascicolo 14 a pag. 19 modifica così:**

## Sintassi del comando

CAT [ # canale, ] numero drive

LINGUAGGIO

differenza degli altri comandi del Microdrive, ERASE non richiede l'asterisco. Naturalmente, perché sia possibile cancellarlo, il file deve essere presente sulla cartuccia; tentativi di cancellare file inesistenti porteranno inevitabilmente al messaggio:

FILE NOT FOUND

(file inesistente)

L'operazione di cancellazione va sempre fatta prima di aggiornare un qualsiasi programma. Se infatti cercherai di memorizzare una nuova versione di PIPPO senza aver precedentemente cancellato quella vecchia, otterrai un errore del tipo:

WRITING TO A "READ" FILE

(stai cercando di scrivere su un file di "LETTURA")

Così, tutte le volte che vorrai registrare un programma il cui nome è già presente nella cartuccia avrai due possibilità: o cambiare il nome del file da registrare, oppure cancellare preventivamente il vecchio file dal Microdrive.

**Sintassi del comando**

ERASE "m"; numero drive; "nome"

25

**Fascicolo 14 a pag. 25 modifica togliendo la parte segnata in rosso**

LINGUAGGIO

INPUT) e riceve in ingresso i dati dalla tastiera. Il flusso 2 è invece in comunicazione con il canale "S", e stampa i caratteri sulla parte superiore dello schermo; il flusso 3 è connesso al canale "P", ed invia le informazioni alla stampante.
Non è possibile utilizzare i flussi 2 e 3 per l'input: qualsiasi tentativo di leggere qualcosa, per esempio dal flusso 3, produrrà il messaggio di errore

INVALID I/O DEVICE

Il simbolo del flusso è il cancelletto (#).
Per il momento faremo riferimento allo Spectrum non ancora connesso al Microdrive e all'interfaccia 1.

**OPEN/CLOSE**

Il concetto di flusso è abbastanza semplice da comprendere; può tuttavia darsi che tu ti stia domandando come fare e quale utilità possa avere associare dei numeri di flusso con dei dispositivi di ingresso/uscita. Avrai la risposta entro poco tempo. Innanzitutto diciamo che l'unità centrale, prima di poter dialogare con una periferica attraverso un flusso deve aprire (OPEN) questo flusso. L'apertura del flusso assolve quindi a un duplice scopo.
Innanzitutto associa un numero di flusso a un particolare dispositivo di I/O; in secondo luogo segnala il dispositivo di I/O che dovrà essere utilizzato.
Per aprire un flusso il BASIC dello Spectrum fornisce il comando

OPEN # F, C

dove "F" è il numero del flusso da aprire, mentre "C" è la stringa che specifica il canale da associare a tale flusso. Una volta impartito questo comando qualsiasi dato inviato o letto attraverso il flusso appena aperto si riferirà al canale "C".
Così

OPEN # 5, "K"

aprirà il flusso 5, associando ad esso la tastiera.
Adesso, se impartiamo questo comando:

INPUT # 5; A, B

potremo assegnare i valori alle variabili A e B nella stessa maniera di una normale istruzione INPUT, visto che il flusso è stato aperto verso la tastiera (KEYBOARD).

10

**Fascicolo 14 a pag. 10 modifica togliendo la parte segnata in rosso**

# MODIFICHE

**Fascicolo 14 a pag. 27 modifica così la linea 30 del listato:**

```
30 INPUT # 4; A
```

**Fascicolo 16 a pag. 24 modifica così la linea 300 del listato:**

```
300 LET A= FN M (C, S)
```

## LINGUAGGIO

un valore dall'area di stack e riporta l'esecuzione del programma all'indirizzo corrispondente al valore prelevato.
Terminata finalmente la parte teorica passiamo adesso a quella pratica, cioè all'applicazione dei concetti visti finora.
Cominciamo subito con un esempio facile, facile....: supponiamo di voler scrivere un carattere (per esempio una "A") nell'angolo in alto a sinistra dello schermo. Sappiamo già come farlo in BASIC: basta una PRINT AT e il gioco è fatto. Adesso dobbiamo però dimenticarci di queste comodità e cercare di fare tutto con i nostri mezzi. Bisogna innanzitutto ricordarsi il meccanismo per cui i caratteri possono apparire sullo schermo: ciascuna posizione dello schermo è infatti composta da una "griglia" di 8x8 quadratini. Ogni gruppo di 8 quadratini disposti su una stessa riga corrisponde a un byte; ciascun carattere viene descritto complutamente da 8 byte. Per visualizzare sullo schermo un certo carattere basterà quindi memorizzare nelle 8 locazioni corrispondenti alla posizione prescelta gli 8 valori che lo identificano. Dal momento che la memoria video comincia alla locazione 16384, per visualizzare la nostra A in BASIC potremo fare

```
POKE 16384,0
POKE 16640,60
POKE 16896,66
POKE 17152,66
POKE 17408,126
POKE 17664,66
POKE 17920,66
POKE 18176,0
```

Eseguendo queste istruzioni in modo immediato, vedremo apparire, un pezzetto alla volta, il nostro carattere A. Esaminiamo adesso come fare in linguaggio macchina. Il procedimento non cambierà di molto: bisognerà caricare in memoria i vari valori:

```
LD A,0
LD (4000H),A
LD A,60
LD (4100H),A
LD A,66
LD (4200H),A
LD (4300H),A
LD A,126
LD (4400H),A
LD A,66
LD (4500H),A
LD (4600H),A
LD A,0
LD (4700H),A
```

Carichiamo il linguaggio macchina utilizzando il solito programma BASIC:

```
10 CLS:RESTORE:CLEAR 32500
20 LET INIZIO=32500:LET FINE=33000
30 FOR X=INIZIO TO FINE
40 READ A
50 IF A=999 THEN GO TO 80
60 POKE X,A
70 NEXT X
80 RANDOMIZE USR 32500:STOP
90 DATA...qui venno scritti i codici
```

Per inserire i codici del programma in linguaggio macchina nella linea

23

**Fascicolo 19 a pag. 23 modifica così:**

```
10 CLS : RESTORE : CLEAR 32499
```